AMENDMENTS TO THE SPECIFICATION

In the Specification:

Please replace the paragraph beginning on page 5, line 19, with the following amended paragraph:

The RSD language component 102 also facilitates moving (or mapping) data between the relational database 100 and an Object component 106 and/or an XML component 108 using a mapping component 110. This is accommodated by using a declarative means rather than a conventional procedural mechanism (*e.g.*, executing C++ code against a result set abstraction to generate an object or an XML structure/component). The capability to map data from one data model to a different data model is a desirable operation in great demand with data environments of today. That is, data environments that are diverse, and employ a wide range of mechanisms and mediums for persisting and accessing data. With respect to Object data, XML data, and Relational databases, the means to map data between each of these different data structures is important, since users are continually modifying their data storage schemas, mediums, and processes.

Please replace the paragraph beginning on page 5, line 30, with the following amended paragraph:

Thus, there is provided the relational database 100 having a relational schema therein represented in the form of metadata, and from which the metadata can be retrieved by the RSD component 102 for generating the RSD file 104. The RSD language component 102 prepares the database data for mapping to another data model *via* the mapping component [[104]] 110. The mapping component [[104]] 110 can then map the data to at least the Object component 106 and/or the XML component 108. Note, however, that the RSD component 102 can be used in conjunction with the mapping component [[104]] 110 to map relational data to an arbitrary domain.

Please replace the paragraph beginning on page 6, line 7, with the following amended paragraph:

The use of an Object Schema Definition (OSD) language component 112 to process Object data 114 for use by the XML component 108 and a Relational component 116, and an XSD language component 118 to process XML data 120 for use by the Object component 106 and the Relational component 116, are not part of this description, as indicated by dotted lines. Note that the particular source data (100, 114, and 120) and the associated language (102, 112, and 118) are not restricted to data transformation to a different target component (106, 108, and 116). That is, a scenario can exist where the source relational database [[110]] 100 can be mapped to a target relational database (the relational component 116) having a different relational structure. Thus, the RSD language component 102 and mapping component 110 facilitate this process. Similarly, a scenario can exist where the source object data 114 can be mapped to a target object data model (the object component 106) having a different object structure. The OSD language component 112 and mapping component 110 facilitate this process. Still further, a scenario can exist where the source XML data 120 can be mapped to a target XML data model (the XML component 108) having a different XML structure. The XSD language component 118 and mapping component 110 facilitate this process.

Please replace the paragraph beginning on page 7, line 3, with the following amended paragraph:

Flow begins at 200 where a tool of the disclosed architecture is activated according to trigger data. Some types of triggering events are described hereinbelow. At 202, the tool, as part of the RSD component, executes to walk through the relational database metadata to find the tables and columns and[[,]] relationships therebetween. At 204, the tool provides the capability to allow the user to select all or a subset of the relationships for use in the RSD file. At 206, the tool allows the user to make a selection. At 208, the tool creates the RSD file of the selected relationships that precisely describe the database structure and data. At this point, optional extensions are included to support implementation-specific extensions and derivations from an ANSI (American National Standards Institute) standard schema (*e.g.*, SQL Server, Oracle[[…]]). At 210, the RSD file is stored for later access. The process then reaches a Stop block.

Please replace the paragraph beginning on page 8, line 1, with the following amended paragraph:

A database is typically defined at least according to tables and columns. The relationships between tables [[is]] <u>are</u> not well-defined, which are the logical components of the relational database. The logical elements are useful for representing the semantics of the database, for mapping the database to another data model, for modeling, etc. A way to obtain a "hint" at the table relationships is *via* a foreign key. In order to describe a relationship between tables, the logical element is used. Thus, given an RSD file, the relational database is recreated by using both the physical and logical elements. The tool 300 is sufficiently sophisticated to handle merge scenarios where the RSD file 104 has been updated by the user with logical information and is then refreshed from the database 100.

Please replace the paragraph beginning on page 9, line 6, with the following amended paragraph:

In another scenario, if certain portions of the database 100 are determined to be a higher priority data [[then]] than other portions, then after changes have been made in the higher priority data, the tool could be automatically activated to update the RSD file 104.

Please replace the paragraph beginning on page 9, line 12, with the following amended paragraph:

In FIG. 3, the RSD file 104 is illustrated local to the RSD component 102. This represents that the RSD file 104 can be generated and stored local to the RSD component 102, where the RSD component 102 is local to the database 100. Alternatively, the RSD file 104 can later be moved to the database 100 where the RSD component 102 is remote from the database 100. All that is important[[,]] is that the file 104 is located where it can be readily accessed or provided to a user who needs access thereto.

Please replace the paragraph beginning on page 10, line 6, with the following amended paragraph:

For example, a support vector machine (SVM) classifier can be employed. An SVM operates by finding a hypersurface in the space of possible inputs. This hypersurface will attempt to split the triggering criteria from the non-triggering events. Intuitively, this makes the classification correct for testing data that is near, but not identical to the training data. Other directed and undirected model classification approaches, ~~include~~ including, *e.g.*, naïve Bayes, Bayesian networks, decision trees, and probabilistic classification models providing different patterns of independence, can be employed. Classification as used herein also is inclusive of statistical regression that is utilized to develop models of priority.

Please replace the paragraph beginning on page 10, line 26, with the following amended paragraph:

Referring now to FIG. 5, there is illustrated a block diagram of a system where the RSD component 102 (or 400 as referenced in FIG. 4) is located remote to the relational database from at least one relational database 100. The database 100 and the RSD component [[104]] 102 are disposed in communication on a network 502, *e.g.*, the Internet. Thus, the RSD component [[104]] 102 can be employed to remotely connect to the database 100 and perform generation of the RSD file that represents the structure and data thereof. The RSD file can then be stored local to the database 100 for access and use for various purposes, as described hereinabove.

Please replace the paragraph beginning on page 31, line 8, with the following amended paragraph:

Precision is only defined for the decimal types: numeric, decimal, float, real. Scale is only defined for the exact decimal types: numeric, decimal.

Please replace the paragraph beginning on page 33, line 22, with the following amended paragraph:

This element serves as a container for the primary key, and the alternate keys[[.]], as well as that which can be represented by UniqueConstraints in the database or constraints on a particular table.